FreshMem: Brain-Inspired Frequency-Space Hybrid Memory for Streaming Video Understanding

Kangcong Li[1], **Peng Ye**[2,3*], **Lin Zhang**[1], **Chao Wang**[1],
Huafeng Qin[4], **Jiayuan Fan**[1,*], **Tao Chen**[1,5]
[1]Fudan University [2]Shanghai AI Lab [3]CUHK
[4]Chongqing Technology and Business University
[5]Shanghai Innovation Institute

Abstract

Transitioning Multimodal Large Language Models (MLLMs) from offline to online streaming video understanding is essential for continuous perception. However, existing methods lack flexible adaptivity, leading to irreversible detail loss and context fragmentation. To resolve this, we propose ***FreshMem***, a **Fre**quency-**S**pace **H**ybrid **Mem**ory network inspired by the brain's logarithmic perception and memory consolidation. FreshMem reconciles short-term fidelity with long-term coherence through two synergistic modules: Multi-scale Frequency Memory (MFM), which projects overflowing frames into representative frequency coefficients, complementing by residual details to reconstruct a global historical "gist"; and Space Thumbnail Memory (STM), which discretizes the continuous stream into episodic clusters by employing an adaptive compression strategy to distill them into high-density space thumbnails. Extensive experiments show that FreshMem significantly boosts the Qwen2-VL baseline, yielding gains of 5.20%, 4.52%, and 2.34% on StreamingBench, OV-Bench, and OVO-Bench, respectively. Besides its plug-and-play design, following low-cost component-wise fine-tuning, FreshMem outperforms current fully fine-tuned methods and achieves a state-of-the-art performance of 82.31% on StreamingBench, exceeding the baseline by 13.31%, offering a highly efficient paradigm for long-horizon streaming video understanding.

1 Introduction

Multimodal Large Language Models (MLLMs) [53, 1, 48, 8] have witnessed remarkable progress in long video understanding [29, 62, 42], demonstrating sophisticated comprehension and reasoning capabilities across diverse tasks. However, most existing methods predominantly operate under an offline paradigm [43, 15], analyzing video sequences as static, complete entities with a priori access to the entire temporal context. As MLLMs gravitate toward more dynamic and interactive environments, there is an urgent need to transition from such post-hoc analysis to online streaming video understanding [7, 18]. This shift represents a fundamental evolution toward continuous perception, requiring models to reason over information as it unfolds under strict causal constraints. Thus, a key challenge emerges in maintaining a delicate balance between long-term information preservation and the seamless integration of novel, incoming frames, which is vital for MLLMs to achieve robust streaming video understanding.

Existing works have primarily evolved along two trajectories: input-level modulation [52, 6], which manages data influx by merging or sparsifying frames, and memory-augmented architectures, which utilize mechanisms like KV cache eviction [10, 34], global memory tokens condensation [37, 59], or external memory bank construction [50, 60]. While these methods mitigate computational pressure,

*Corresponding authors: jyfan@fudan.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

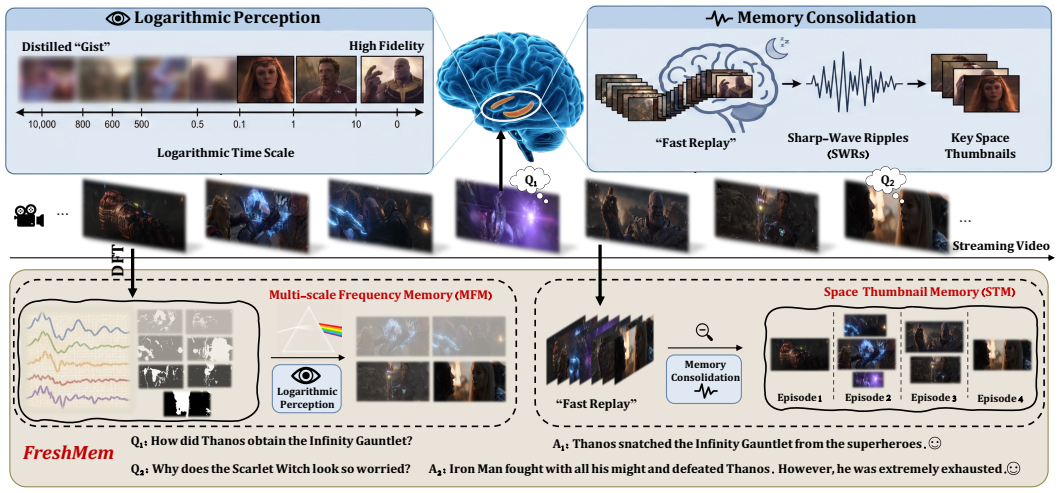


Figure 1: **Overview of FreshMem and its biological foundations.** (Top-Left) Logarithmic Perception, where the brain maintains high fidelity for immediate stimuli while distilling the distant past into a semantic "gist"; (Top-Right) Memory Consolidation, where SWRs compress continuous experiences into key space thumbnails; (Bottom) FreshMem materializes these principles via two synergistic modules: MFM and STM. This hybrid design enables the model to effectively reconcile short-term fidelity with long-term coherence for streaming video understanding.

they often precipitate an irrevocable loss of fine-grained details or struggle to reconcile short-term temporal fidelity with long-term contextual coherence. A fundamental limitation remains in their uniform treatment of temporal dynamics, which relies on rigid heuristics or static downsampling. Consequently, current frameworks lack the flexibility to adaptively adjust representation density over time. As the temporal horizon extends, insufficient memory consolidation leads to fragmented representations, compromising action continuity and semantic consistency, and ultimately limiting the model's ability to sustain a coherent long-horizon narrative.

To address these structural deficiencies, we draw inspiration from the brain's memory mechanisms, which effectively reconcile above bottleneck via two core mechanisms: logarithmic perception [12, 40] for temporal organization and memory consolidation [11, 4] for information refinement. Neuroscience research [40] indicates that human brain encodes time on a non-uniform, logarithmic scale, preserving immediate stimuli with high fidelity while progressively distilling the distant past into a semantic "gist". This suggests that a non-uniform temporal representation is fundamental for managing both local detail and global context. Complementing this, Neuroscience evidence [4] reveals that the brain undergoes memory consolidation via mechanisms such as Sharp-Wave Ripples (SWRs), where extensive experiences are condensed into concise space thumbnails. This rapid replay and distillation process ensures the stability of long-term semantic structures. By emulating these dual principles, a streaming model can theoretically enable a more adaptive approach to maintaining continuous representations over extended horizons.

To translate these biological insights into a deployable architecture, we propose ***FreshMem***, a **Fre**quency-**S**pace **H**ybrid **Mem**ory network. To embody logarithmic perception, we introduce the Multi-scale Frequency Memory (MFM) module. Rather than the uniform discarding prevalent in existing models, MFM utilizes Discrete Fourier Transforms (DFT) [25, 13, 14, 5] to project historical frames overflowing the sliding window [52] into representative frequency coefficients. By using the superposition of these frequencies, MFM reconstructs the global "gist" of past context with minimal overhead, mitigating the raw data redundancy described earlier. Parallelly, the memory consolidation mechanism is materialized through a Space Thumbnail Memory (STM) module. By monitoring episodic boundaries via cosine similarity, STM discretizes the continuous stream into distinct episodic clusters. An adaptive compression strategy then transforms these clusters into high-density space thumbnails, utilizing a centroid-based fusion mechanism to merge episodes with high semantic affinity.

To validate the effectiveness of the proposed FreshMem, we conduct extensive evaluations across both online streaming and offline video understanding benchmarks. In streaming scenarios, FreshMem delivers substantial performance gains over the competitive Qwen2-VL [48] baseline, achieving absolute accuracy improvements of 5.20% on StreamingBench [30], 4.52% on OV-Bench [18], and 2.34% on OVO-Bench [35]. Notably, even as a training-free solution, FreshMem outperforms several fully fine-tuned counterparts, underscoring its superior capacity for handling continuous data streams. Furthermore, despite its plug-and-play design, FreshMem can be further enhanced through low-cost component-wise fine-tuning. This configuration achieves a state-of-the-art performance of 82.31% on StreamingBench, representing a significant 13.31% absolute improvement over the baseline and surpassing existing fully fine-tuned methods. Beyond streaming tasks, FreshMem also maintains robust performance on traditional offline benchmarks, yielding consistent improvements on the long-video benchmark MLVU [64] and the short-video benchmark MVBench [26]. Overall, these results demonstrate that FreshMem effectively reconciles short-term fidelity with long-term coherence, offering a highly efficient and scalable alternative to computationally intensive paradigms. The main contributions of our work are summarized as follows:

Brain-inspired Paradigm for Streaming Video. We bridge the gap between biological memory mechanisms and streaming MLLMs by mimicking the human brain's logarithmic perception and memory consolidation. This novel perspective effectively addresses the structural dilemma between maintaining short-term fidelity and ensuring long-term semantic coherence in continuous video streams.

Frequency-Space Hybrid Memory Network. We propose FreshMem, a unified framework comprising two core components: the MFM module, which projects overflowing frames into representative frequency coefficients complemented by salient residuals to reconstruct the global "gist" of historical context; and the STM module, which discretizes continuous streams into episodic clusters and employs an adaptive compression strategy to distill them into high-density space thumbnails.

Training-free Versatility with Scalable SOTA Performance. As a plug-and-play module, FreshMem delivers immediate training-free gains that surpass many fully fine-tuned methods. Furthermore, its performance can be further amplified through low-cost component-wise fine-tuning to achieve state-of-the-art results, offering a highly efficient and scalable paradigm for streaming video understanding.

2 Related Work

2.1 Input-level Modulation for Video Understanding

In the context of long-form video, static downsampling remains the dominant paradigm. Methods like AKS [44], and BOLT [32] employ heuristic metrics or learnable policies to filter redundant frames based on query relevance, while TimeViper [51] adopts linear-complexity architectures to process denser inputs efficiently. In dynamic streaming scenarios, modulation strategies shift toward managing a sliding computational window. TimeChat [38] attends time information by a time-aware frame encoder and a sliding window Q-Former, while StreamingAssistant [21] further reduces redundancy by pruning tokens based on spatial adjacency or decoupling positional encodings.

Current methods, limited by rigid spatial heuristics and uniform windowing, overlook temporal nuances. Consequently, they sacrifice high-frequency details like subtle motion, hindering the reconciliation of local fidelity with the evolving narrative.

2.2 Memory-augmented Architectures for Video Understanding

A primary direction involves KV cache compression. LiveVLM [34] and InfiniPot-V [22] dynamically prune redundant key-value pairs in the streaming cache. Another direction focuses on global token condensation and external banks. Pioneering works like MovieChat [43] and MA-LMM [15] condense history into token-based summaries. More recent frameworks enforce structure upon these memories: Dispider [36] features a lightweight proactive streaming video processing module to determine the use of memory, and Flash-VStream [60] employs dual-layer storage. Furthermore, VideoLLaMB [49] introduces recurrent bridges to smooth transitions.

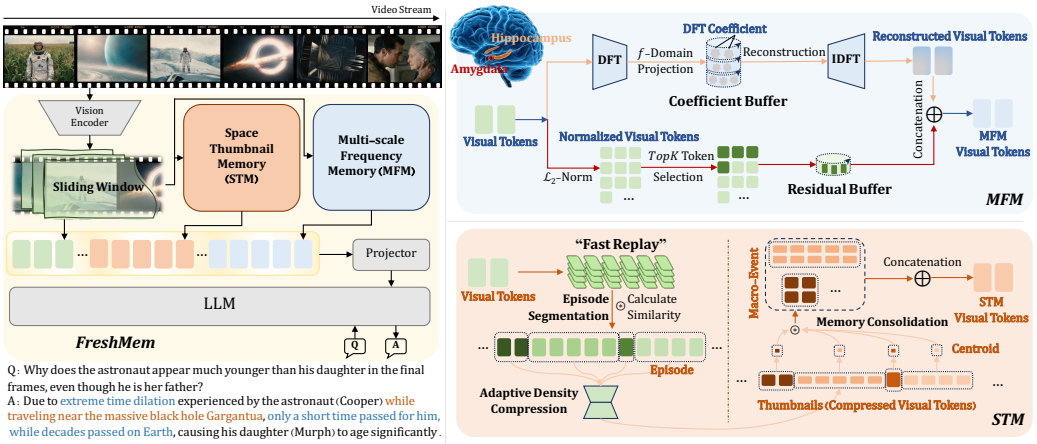


Figure 2: **Illustration of FreshMem.** (1) Sliding window for immediate high-fidelity context; (2) MFM, which maintains a global history by frequency coefficients and salient residuals; and (3) STM, which performs online episode segmentation and adaptive compression to keep key spatial thumbnails. The concatenated memory features enable the LLM to answer long-horizon queries with high accuracy.

Current architectures fail to capture continuous temporal dynamics, often yielding fragmented representations caused by heavy spatial or lossy textual dependencies. Our proposed FreshMem overcomes this by balancing the retention of critical long-term information with short-term sensitivity, ensuring robust long-horizon reasoning.

3 *FreshMem*

3.1 Theory Foundations

FreshMem is grounded in two fundamental neuroscientific observations regarding temporal perception and episodic replay: the Weber-Fechner Law [12] of logarithmic perception [40] and the Sharp-Wave Ripples (SWRs) [4] mechanism for memory consolidation [11].

Weber-Fechner Law. Research in psychophysics suggests that the brain's perception of time is non-uniform [40]. According to the Weber-Fechner Law, the relationship between the objective magnitude of a physical stimulus and its subjective perception is logarithmic rather than linear [20].

Formally, let Δt be the time interval from the current moment. The perceived information density $I(\Delta t)$ can be modeled as:

$$I(\Delta t) \propto \frac{1}{k \cdot \ln(\Delta t + \epsilon)} \tag{1}$$

where k is a scaling constant and ϵ prevents singularity. This biological prior motivates us to project historical context into a domain where information density naturally decays, rather than storing frames with uniform density.

Sharp-Wave Ripples (SWRs). While the Weber-Fechner law describes passive decay, the brain also employs active consolidation mechanisms. During slow-wave sleep or rest, the hippocampus generates Sharp-Wave Ripples (SWRs). These high-frequency oscillations support the "fast replay" of sequential experiences, compressing long continuous episodes into brief, high-density neural bursts [4].

Mathematically, we model this biological process as a Space Compression Operator Ψ. Given a continuous stream of visual features $E = \{x_1, x_2, ..., x_T\}$ representing an episode, the SWRs mechanism functions as a non-linear integration that maps the sequence to a high-density "memory trace" $\mathbf{m}$:

$$\mathbf{m} = \Psi(E) \approx \frac{1}{|E|} \sum_{i=1}^{T} \mathbb{I}(x_i) \cdot x_i \tag{2}$$

where $\mathbb{I}(\cdot)$ denotes a saliency filter. This implies that extensive space redundancy is removed, retaining only the topological "centroid" of the event. This inspires our strategy of dynamically monitoring semantic boundaries to fuse redundant frames into "episodic clusters", preserving topology within a limited budget.

3.2 Frequency-Space Hybrid Memory Network

Guided by these principles, we propose the FreshMem framework. As illustrated in the left of Figure 2, video frames are processed by a vision encoder and buffered in a sliding window $\mathcal{M}_{short}$ for immediate context. To retain long-term information, overflowed frames are parallelly assimilated by Multi-scale Frequency Memory (MFM) for global "gist" reconstruction and Space Thumbnail Memory (STM) for episodic clustering. Finally, tokens from these memories are aggregated via a projector and fed into the LLM, enabling robust long-horizon streaming reasoning.

3.2.1 Multi-scale Frequency Memory (MFM)

To balance immediate precision with long-term retention, MFM operates in conjunction with sliding window that stores raw features of the most recent frames. As frames overflow, they are not discarded but projected into the frequency domain.

Incremental Frequency Update. Standard DFT [25, 13, 14, 5] requires access to the entire history, which is infeasible for streaming. We implement a recurrent update mechanism. Let $C_t \in \mathbb{C}^{K \times D}$ denote the frequency coefficients at step t, where K is the number of frequency bands. For an incoming overflowed frame x_t, we update the coefficients incrementally:

$$C_t[k] = \gamma \cdot C_{t-1}[k] + x_t \cdot e^{-j\omega_k t} \tag{3}$$

where ω_k is the angular frequency derived from the Weber-Fechner scale, and $\gamma \in (0, 1)$ is a decay factor preventing saturation. This ensures $O(1)$ update complexity. The coefficients are stored in a circular buffer structure; when the buffer is full, the pointer wraps around, overwriting the oldest frequency interactions, naturally mimicking the fading of very distant memories [33].

Amygdala-Inspired Residuals. Biological memory is not purely semantic; the *Amygdala* ensures that emotionally salient or shocking events are retained with high fidelity [23], regardless of their frequency characteristics. Inspired by this, and leveraging observations in [46] that feature magnitude correlates with information importance, we introduce a *salient residual mechanism*.

During the update, we use the L2 norm of pre-LLM visual features as a lightweight, magnitude-based intrinsic saliency score for residual token selection, denoted as $\mathcal{B}_{residual}$. We retain the top-k elements with the highest norm as residual tokens $\mathbf{r}_t$:

$$\mathbf{r}_t = \begin{cases} x_t, & \text{if } \|x_t\|_2 \in \text{TopK}(\mathcal{B}_{residual}) \\ \mathbf{0}, & \text{otherwise} \end{cases} \tag{4}$$

This dual-pathway storage mimics the Hippocampus' global low-frequency context and Amygdala's sparse high-frequency saliency collaboration [39].

Holographic Reconstruction. During retrieval at step τ, we reconstruct the history $\hat{H}$ by performing the Inverse Discrete Fourier Transform (IDFT) on the frequency coefficients and fusing them with the retrieved residuals:

$$\hat{H}_\tau = \text{Real}\left(\sum_{k=1}^{K} C_t[k] \cdot e^{j\omega_k \tau}\right) \oplus \mathbf{r}_\tau \tag{5}$$

This reconstruction allows the model to "hallucinate" the gist of the entire history while preserving sharp details for key frames.

3.2.2 Space Thumbnail Memory (STM)

While MFM handles global context, STM is designed to discretize continuous streams into structured "episodic clusters", capturing the topological evolution of events.

Table 1: **Evaluation results on OV-Bench.** FP, THV, PM, SP, STP, TP represent Future Prediction, Temporal Hallucination Verification, Past Memory, Spatial, Spatio-Temporal, Temporal Perception respectively.

Method	FP	THV	PM	SP	STP	TP	**Avg.**
Proprietary MLLMs							
Gemini-1.5-Flash [45]	48.97	52.63	49.03	34.90	68.50	50.83	50.7
Open-source Offline Video MLLMs							
InternVL2-7B [8]	46.80	56.00	54.73	34.70	62.20	37.57	48.7
InternVL2-4B [8]	43.03	56.20	46.13	25.50	57.65	34.50	44.1
LLaMA-VID-7B [29]	38.03	52.77	43.17	21.55	50.25	41.70	41.9
LLaVA-Onevision-7B [24]	55.53	51.73	49.37	28.65	68.10	43.00	49.5
LongVA-7B [62]	50.03	47.20	49.53	17.50	56.05	36.90	43.6
MiniCPM-V2.6-7B [54]	28.07	55.03	34.70	19.20	67.30	32.90	39.1
LITA-7B [17]	21.20	38.20	12.27	10.75	29.55	10.43	20.4
TimeChat-7B [38]	13.90	16.00	11.20	13.50	12.10	10.07	12.8
VTimeLLM-7B [16]	25.20	53.93	32.40	23.15	23.85	33.50	33.1
Open-source Online Video MLLMs							
VideoLLM-Online-7B [7]	7.57	14.57	9.67	13.60	10.40	3.63	9.6
MovieChat-7B [43]	24.73	47.53	26.87	33.25	26.65	25.57	30.9
Flash-Vstream-7B [60]	29.47	47.33	28.27	26.75	24.20	27.40	31.2
InternVL2-2B [8]	44.36	49.66	50.24	23.64	52.72	26.44	42.06
+FreshMem	46.62	54.13	59.69	30.64	57.79	29.76	**47.44**(↑ 5.38%)
Qwen2VL-7B [48]	45.39	50.47	57.80	22.71	69.02	30.66	46.30
+FreshMem	50.38	59.73	60.87	25.51	72.64	32.39	**50.82**(↑ 4.52%)

Table 2: **Evaluation on OVO-Bench.** RTVP, BT, FAR denote Real-Time Visual Perception, Backward Tracing, Forward Active Responding.

Method	RTVP	BT	FAR	**Avg.**
Proprietary MLLMs				
Gemini 1.5 Pro [45]	69.32	62.54	57.15	63.00
GPT-4o [19]	64.46	60.75	53.40	59.54
Open-source Offline Video MLLMs				
LLaVA-Video [63]	63.34	41.68	54.17	53.06
LLaVA-OneVision [24]	62.79	44.99	50.85	52.88
InternVL2-8B [8]	60.73	44.00	45.42	50.05
LongVU-7B [42]	57.40	39.49	48.54	48.48
Open-source Online Video MLLMs				
VideoLLM-online [7]	20.79	17.73	-	12.84
Flash-VStream [60]	29.86	25.35	44.23	33.15
Dispider-7B [36]	54.55	36.06	34.72	41.78
InternVL2-2B [8]	44.92	34.55	41.93	41.22
+FreshMem	47.79	40.10	42.63	**43.53**(↑ 2.31%)
Qwen2VL-7B [48]	60.65	49.13	48.86	52.19
+FreshMem	66.67	54.04	48.25	**54.53**(↑ 2.34%)

Episode Segmentation. Similar to MFM, frames first pass through sliding window. Upon eviction, STM monitors the semantic continuity. We define an episode boundary based on the cosine similarity between the current frame x_t and the preceding context x_{t-1}:

$$\delta(t) = \frac{x_t \cdot x_{t-1}}{|x_t||x_{t-1}|} < \theta_{event} \tag{6}$$

If the similarity score $\delta(t)$ drops below the threshold θ_{event}, a new episodic cluster E_{new} is initialized; otherwise, x_t is appended to the current active episode.

Adaptive Density Compression. Within each episode, we apply an adaptive compression strategy to maintain a constant "information density". The sampling rate ρ is dynamically adjusted based on the episode duration N. For an episode E_i with N frames, the stored representation $\mathbf{Z}_i$ is obtained by adaptive downsampling:

$$\mathbf{Z}_i = \text{Sample}(E_i, \rho(N)) \tag{7}$$

where the sampling rate $\rho(N) \propto \text{clamp}(N, \rho_{min}, \rho_{max})$ adapts to the episode length. This ensures that short, fast-paced actions retain higher frame rates, while long, redundant stationary scenes are heavily compressed, optimizing the storage budget.

Centroid-based Memory Consolidation. To strictly enforce the memory budget, we propose a topology-aware merge strategy. Instead of simply discarding the oldest event (FIFO), we calculate the *semantic centroid* [41] μ_i for each stored episode E_i. When the memory is full, we compute the pairwise similarity S_{merge} between adjacent event centroids:

$$S_{merge}(i, i+1) = \frac{\mu_i \cdot \mu_{i+1}}{|\mu_i||\mu_{i+1}|} \tag{8}$$

If S_{merge} exceeds the fusion threshold θ_{merge}, the two episodes are merged into a single "Macro-Event", summing their frame counts and re-calculating the unified centroid. This mechanism effectively fuses redundant micro-events into a coherent narrative, preserving the deep semantic topology of the video.

3.2.3 Global Memory Integration

Finally, the comprehensive memory representation $\mathcal{M}_{total}$ for the FreshMem framework is constructed by concatenating the outputs of all three modules:

$$\mathcal{M}_{total} = [\mathcal{M}_{MFM} \parallel \mathcal{M}_{STM} \parallel \mathcal{M}_{short}] \tag{9}$$

This hybrid token sequence provides the downstream predictor with global historical context, structured episodic summaries and immediate details.

Table 3: **Evaluation results on StreamingBench.**

Method	OP	CR	CS	ATP	EU	TR	PR	SU	ACP	CT	ALL
Proprietary MLLMs											
Gemini 1.5 Pro [45]	79.02	80.47	83.54	79.67	80.00	84.74	77.78	64.23	71.95	48.70	75.69
GPT-4o [19]	77.11	80.47	83.91	76.47	70.19	83.80	66.67	62.19	69.12	49.22	73.28
Claude 3.5 Sonnet [2]	73.33	80.47	84.09	82.02	75.39	79.53	61.11	61.79	69.32	43.09	72.44
Open-source Offline Video MLLMs											
Video-LLaMA2-7B [9]	55.86	55.47	57.41	58.17	52.80	43.61	39.81	42.68	45.61	35.23	49.52
LongVA-7B [62]	70.03	63.28	61.20	70.92	62.73	59.50	61.11	53.66	54.67	34.72	59.96
InternVL2-8B [8]	68.12	60.94	69.40	77.12	67.70	62.93	59.26	53.25	54.96	56.48	63.72
Kangaroo-7B [31]	71.12	84.38	70.66	73.20	67.08	61.68	56.48	55.69	62.04	38.86	64.60
LLaVA-NeXT-Video-32B [63]	78.20	70.31	73.82	76.80	63.35	69.78	57.41	56.10	64.31	38.86	66.96
MiniCPM-V2.6-8B [54]	71.93	71.09	77.92	75.82	64.60	65.73	70.37	56.10	62.32	53.37	67.44
LLaVA-OneVision-7B [24]	80.38	74.22	76.03	80.72	72.67	71.65	67.59	65.45	65.72	45.08	71.12
Qwen2.5-VL-7B [3]	78.32	80.47	78.86	80.45	76.73	78.50	79.63	63.41	66.19	53.19	73.68
Open-source Online Video MLLMs											
Flash-VStream-7B [60]	25.89	43.57	24.91	23.87	27.33	13.08	18.52	25.20	23.87	48.70	23.23
VideoLLM-online-8B [7]	39.07	40.06	34.49	31.05	45.96	32.40	31.48	34.16	42.49	27.89	35.99
Dispider-7B [36]	74.92	75.53	74.10	73.08	74.44	59.92	76.14	62.91	62.16	45.80	67.63
ReKV-7B [10]	74.39	78.91	78.55	77.12	68.32	67.91	67.59	62.60	64.31	44.56	69.08
LiveVLM-7B [34]	81.47	78.13	83.28	79.08	69.57	74.14	75.00	69.11	67.71	40.41	72.92
StreamForest [59]	83.11	82.81	82.65	84.26	77.50	78.19	76.85	69.11	75.64	54.40	77.26
InternVL2-2B [8]	58.58	50.78	59.62	66.89	57.50	49.22	52.78	48.37	47.31	18.13	52.08
+FreshMem	64.85	57.81	66.56	72.46	63.75	58.26	60.19	52.44	53.82	19.17	58.21(↑ 6.13%)
+FreshMem(ViT fine-tuned)	62.47	62.13	65.82	69.29	67.54	63.88	66.03	59.71	60.35	24.56	60.44(↑ 8.36%)
+FreshMem(Projector fine-tuned)	63.74	63.29	65.81	70.33	68.57	64.98	67.12	59.46	60.03	25.67	60.82(↑ 8.74%)
+FreshMem(LLM fine-tuned)	62.39	66.12	65.87	70.54	71.63	67.98	69.45	61.26	61.08	27.31	**62.69(↑ 10.61%)**
Qwen2VL-7B [48]	77.38	76.56	73.19	75.08	75.00	67.91	73.15	65.04	66.57	35.75	69.00
+FreshMem	84.47	83.59	77.60	83.28	78.12	80.37	70.37	74.39	66.86	30.05	74.20(↑ 5.20%)
+FreshMem(ViT fine-tuned)	82.24	85.29	78.10	82.25	81.68	78.27	75.85	77.72	73.64	36.54	76.35(↑ 7.35%)
+FreshMem(Projector fine-tuned)	84.93	82.28	80.63	84.26	82.74	78.22	77.18	79.63	74.75	36.47	77.73(↑ 8.73%)
+FreshMem(LLM fine-tuned)	82.65	90.32	90.18	83.79	85.01	85.64	84.36	76.62	83.14	58.48	**82.31(↑ 13.31%)**

4 Experiments

We employ Qwen2-VL-7B [48] and InternVL2-2B [8] as baselines and test the performance on both online benchmarks (see Section 4.1) and offline benchmarks (see Section 4.2). FreshMem is training-free and it can be directly applied on top of baseline video MLLMs during inference. All experiments are conducted on RTX4090 and H200 GPUs. Detailed settings for all parameters are provided in Table 7.

4.1 Online Benchmark Results

We evaluate the performance of our model on three benchmarks for online video question answering: OV-Bench [18], OVO-Bench [35], and StreamingBench [30]. These benchmarks are formulated as multiple-choice questions. For each question, the answer must be generated solely based on the video content before the corresponding timestamp, without access to the entire video.

Experimental results of these three benchmarks are shown in Tables 1 to 3. FreshMem achieves stable and significant performance improvement over different baselines and benchmarks. Built upon Qwen2-VL-7B, it obtains an accuracy of 50.82% on OV-Bench, 54.53% on OVO-Bench, and 74.2% on StreamingBench, yielding absolute improvements of 4.52%, 2.34%, and 5.20% over the baseline, respectively. This improvement arises from FreshMem's brain-inspired memory strategy, which retains recent details with a frequency domain projection while compressing long-term context into semantic gist and episodic thumbnails. Notably, our method outperforms several supervised approaches, despite not requiring extensive training. By aligning memory retention with video temporal structure, our method preserves both short-term precision and long-term coherence without additional training. More Experiments on InternVL3.5-7B and DeepSeek-VL2-Tiny are in Table 19.

To explore trainable extensions of FreshMem, following the end-to-end training settings established by StreamForest [59], we independently fine-tuned the ViT, the Projector, and the LLM components of our architecture. Results in Table 3 show FreshMem achieves a dominant 82.31%, vastly outperforming both our training-free baseline and the fully trained StreamForest model and reaching state-of-the-art performance.

4.2 Offline Benchmark Results

We further evaluate our method on one long video understanding benchmark MLVU [64] and one short video benchmark MVBench [26] and report the results in Table 4. In these offline settings, the MLLM is given access to the complete video as input. Our method demonstrates improvement upon the Qwen2VL-7B baseline by 2.4% on MLVU and by 4.7% on MVBench. These results highlight that our method not only excels on online benchmarks but also generalizes effectively to offline benchmarks.

Table 4: **Evaluation results on MLVU and MVBench.**

Method	MLVU	MVBench
LongVA-7B [62]	56.3	-
LLaVA-OneVision-7B [24]	64.7	56.7
LLaVA-Video-7B [63]	70.8	58.6
MovieChat-7B [43]	-	55.1
Qwen2VL-7B [48]	54.9	59.1
+FreshMem	**57.3**(↑ 2.4%)	**63.8**(↑ 4.7%)

Table 5: **Ablation study on OVO-Bench results.** SW represents Sliding Window.

Model	SW	STM	MFM	Avg.
Qwen2-VL-7B	×	×	×	52.19
+ SW	✓	×	×	52.52
+ STM	×	✓	×	53.64
+ MFM	×	×	✓	53.54
+ SW + STM	✓	✓	×	53.92
+ SW + MFM	✓	×	✓	54.36
+ STM + MFM	×	✓	✓	54.05
Ours	✓	✓	✓	**54.53**

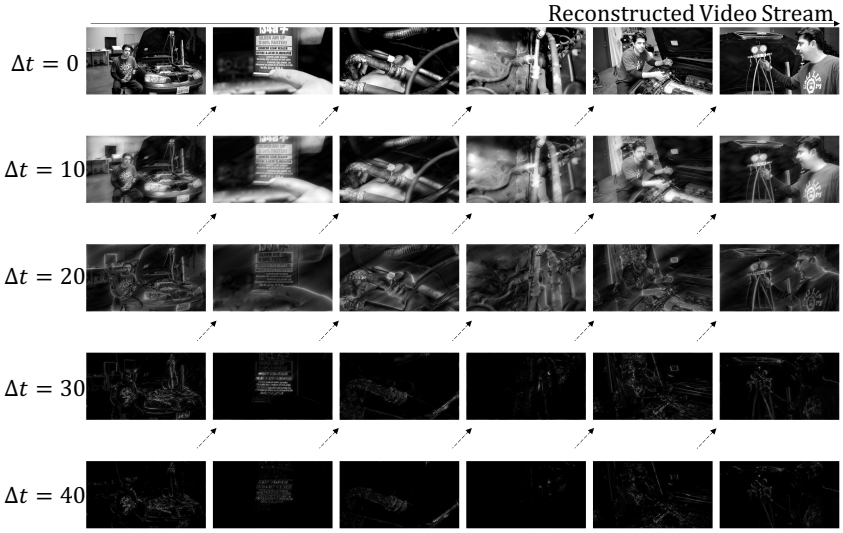


Figure 3: **Reconstructed video stream from frequency coefficients.** The visualization demonstrates the information decay inherent in our MFM module.

4.3 Visualization and Case Study

To enhance the interpretability of FreshMem, we visualized the reconstructed visual tokens during the streaming video process by decoding them back into complete video frames, as illustrated in Figure 3. As observed in the figure, frames further from the current timestamp (Δt) appear increasingly blurred and tend towards a binary-like representation. This demonstrates that following DFT and IDFT processing, the MFM effectively preserves the "gist" information—maintaining a rough outline of past events that gradually decays over time. This phenomenon empirically validates the rationality of the logarithmic perception mechanism inherent in our MFM module.

Furthermore, we visualized the reconstruction of the top-k residual tokens selected within the MFM, as shown in Figure 4. The visualization reveals that the selected top-k tokens are predominantly distributed along the edges of people and objects, clustering in semantically dense regions within each frame. This empirically validates that these residual tokens serve as a vital complement to the

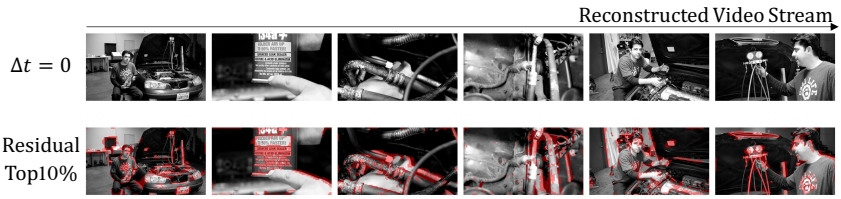


Figure 4: **Impact of Salient Residual Tokens.** We visualize the reconstruction of the top-10% residual tokens selected by the MFM.

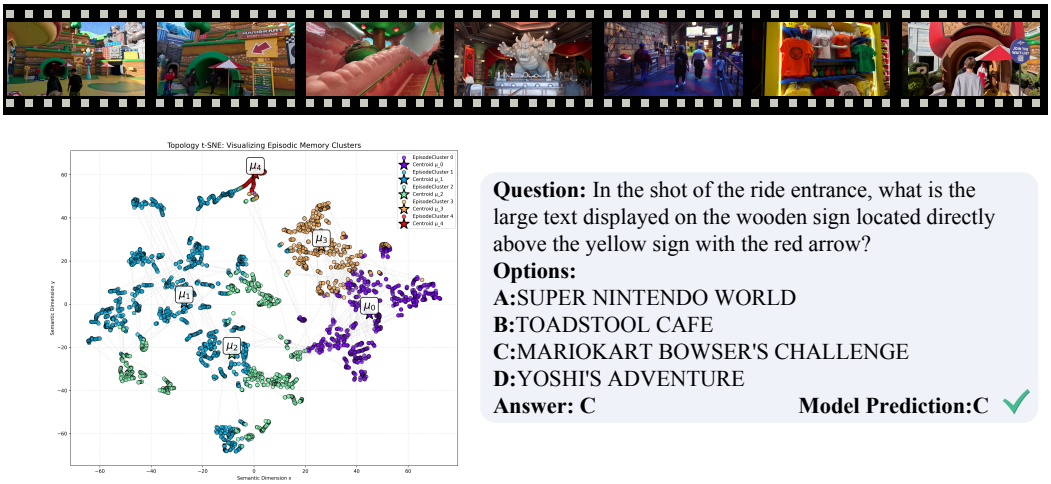


Figure 5: **Interpretability analysis of the STM module.** The t-SNE visualization illustrates the effectiveness of our Memory Consolidation mechanism, showing how continuous video streams are discretized into coherent episodic clusters.

reconstructed visual tokens. Specifically, while the reconstructed visual tokens for timestamps distant from the present tend to fade into a blurred background, the residual tokens effectively delineate sharp contours and structural details on top of them. Together, these two components work synergistically to achieve robust memory and recall of past events.

Finally, we conduct an interpretability experiment and a case study specifically for the STM module, as illustrated in Figure 5. We visualized the "episode clusters" using t-SNE and annotated the centroid (μ) of each cluster. In conjunction with the provided case study, we can observe that although the streaming video contains frames that are semantically similar but temporally discontinuous, our proposed episode segmentation and cluster fusion mechanisms enable the model to effectively capture these underlying temporal correlations. This empirically validates that our STM module mimics the biological process of memory consolidation, effectively reducing the semantic fragmentation of visual information.

4.4 Ablation Studies

To evaluate the effectiveness of each component in our proposed method, we conduct a series of ablation experiments on the OVO-Bench. As shown in Table 5, the baseline Qwen2-VL-7B model achieves an average score of 52.19%. By individually integrating the Sliding Window (SW), STM, and MFM, we observe consistent performance gains of 0.33%, 1.45%, and 1.35% respectively, demonstrating the necessity of each module. Combining these components further boosts the performance, and our full configuration (SW + STM + MFM) achieves the best result of 54.53%, which is a 2.34% absolute improvement over the baseline.

5 Conclusion

In this paper, we proposed FreshMem, a brain-inspired framework that reconciles short-term fidelity with long-term coherence via frequency-domain projection and episodic clustering. Extensive experiments demonstrate that FreshMem achieves distinguished performance on multiple streaming benchmarks. Remarkably, despite being a training-free solution, it surpasses several fully fine-tuned methods, highlighting the superior efficacy of our structured memory design. Following low-cost component-wise fine-tuning, FreshMem provides crucial inspiration for designing more reliable parameterized memory structures and cost-effective training recipes. We firmly believe that FreshMem serves as a vital bridge, transitioning multimodal models from heuristic, non-parametric memory designs toward highly efficient, trainable parametric memory paradigms.

Acknowledgments and Disclosure of Funding

This work is supported by National Natural Science Foundation of China (No. 62571138), National Key R&D Program of China (No. 2026YFE0101200), and National Natural Science Foundation of China (No. 62671175). The computations in this research were performed using the CFFF platform of Fudan University.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Anthropic. Claude 3.5 sonnet. Technical report, URL https://www.anthropic.com/news/claude-3-5-sonnet, 2024.

[3] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] G. Buzsáki. *The brain from inside out*. Oxford University Press, 2021.

[5] X. Cai, C. Sun, Y. Wang, H. Yang, and Y. Wu. Pfgnet: A fully convolutional frequency-guided peripheral gating network for efficient spatiotemporal predictive learning, 2026.

[6] D. Chatterjee, E. Remelli, Y. Song, B. Tekin, A. Mittal, B. Bhatnagar, N. C. Camgoz, S. Hampali, E. Sauser, S. Ma, et al. Streaming videollms for real-time procedural video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22586–22598, 2025.

[7] J. Chen, Z. Lv, S. Wu, K. Q. Lin, C. Song, D. Gao, J.-W. Liu, Z. Gao, D. Mao, and M. Z. Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024.

[8] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[9] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[10] S. Di, Z. Yu, G. Zhang, H. Li, H. Cheng, B. Li, W. He, F. Shu, H. Jiang, et al. Streaming video question-answering with in-context video kv-cache retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025.

[11] Y. Dudai. The neurobiology of consolidations, or, how stable is the engram?. *Annual review of psychology vol. 55 (2004): 51-86. doi:10.1146/annurev.psych.55.090902.142050*, 2004.

[12] G. T. Fechner. *Elements of psychophysics [Elemente der Psychophysik].* Leipzig : Breitkopf und Härtel, 1860.

[13] M. Fontana, M. Spratling, and M. Shi. Faar: Efficient frequency-aware multi-task fine-tuning via automatic rank selection, 2026.

[14] C. Gu, L. Chen, L. Gu, and Y. Fu. Fourier angle alignment for oriented object detection in remote sensing, 2026.

[15] B. He, H. Li, Y. K. Jang, et al. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13504–13514, 2024.

[16] B. Huang, X. Wang, H. Chen, Z. Song, and W. Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280, 2024.

[17] D.-A. Huang, S. Liao, S. Radhakrishnan, H. Yin, P. Molchanov, Z. Yu, and J. Kautz. Lita: Language instructed temporal-localization assistant. In *European Conference on Computer Vision*, pages 202–218. Springer, 2024.

[18] Z. Huang, X. Li, J. Li, J. Wang, X. Zeng, C. Liang, T. Wu, X. Chen, L. Li, and L. Wang. Online video understanding: Ovbench and videochat-online. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3328–3338, 2025.

[19] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[20] J. Jeans. *Science & music*. New York : Dover Publications, 1968.

[21] X. Jin, H. Yu, B. Yu, et al. Streamingassistant: Efficient visual token pruning for accelerating online video understanding. *arXiv preprint arXiv:2512.12560*, 2025.

[22] M. Kim, K. Shim, J. Choi, et al. Infinipot-v: Memory-constrained kv cache compression for streaming video understanding. *arXiv preprint arXiv:2506.15745*, 2025.

[23] J. LeDoux. The amygdala. *Current biology*, 17(20):R868–R874, 2007.

[24] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[25] H. Li, Z. Wu, R. Shao, and Y. Fu. Statistical characteristic-guided denoising for rapid high-resolution transmission electron microscopy imaging, 2026.

[26] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[27] K. Li, P. Ye, C. Tu, L. Zhang, C. Song, J. Wu, T. Yang, Q. Zheng, and T. Chen. PaceLLM: Brain-inspired large language models for long-context understanding. In *Advances in Neural Information Processing Systems*, volume 38, pages 95334–95359, 2025.

[28] K. Li, H. Zhang, S. Hu, S. Tang, and T. Chen. ReTTT: Reflection-driven test-time training via experience-based parametric memory, 2026.

[29] Y. Li, C. Wang, and J. Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

[30] J. Lin, Z. Fang, C. Chen, Z. Wan, F. Luo, P. Li, Y. Liu, and M. Sun. Streamingbench: Assessing the gap for mllms to achieve streaming video understanding. *arXiv preprint arXiv:2411.03628*, 2024.

[31] J. Liu, Y. Wang, H. Ma, X. Wu, X. Ma, X. Wei, J. Jiao, E. Wu, and J. Hu. Kangaroo: A powerful video-language model supporting long-context video input. *arXiv preprint arXiv:2408.15542*, 2024.

[32] S. Liu, C. Zhao, T. Xu, et al. Bolt: Boost large vision-language model without training for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3318–3327, 2025.

[33] J. M. Murre and J. Dros. Replication and analysis of ebbinghaus' forgetting curve. *PloS one*, 10(7):e0120644, 2015.

[34] Z. Ning, G. Liu, Q. Jin, W. Ding, M. Guo, and J. Zhao. Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval. *arXiv preprint arXiv:2505.15269*, 2025.

[35] J. Niu, Y. Li, Z. Miao, C. Ge, Y. Zhou, Q. He, X. Dong, H. Duan, S. Ding, R. Qian, et al. Ovo-bench: How far is your video-llms from real-world online video understanding? In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18902–18913, 2025.

[36] R. Qian, S. Ding, X. Dong, P. Zhang, Y. Zang, Y. Cao, D. Lin, and J. Wang. Dispider: Enabling video llms with active real-time interaction via disentangled perception, decision, and reaction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24045–24055, 2025.

[37] R. Qian, X. Dong, P. Zhang, Y. Zang, S. Ding, D. Lin, and J. Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

[38] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024.

[39] G. Richter-Levin and I. Akirav. Amygdala-hippocampus dynamic interaction in relation to memory. *Molecular neurobiology*, 22(1):11–20, 2000.

[40] G. Scheler. Logarithmic distributions prove that intrinsic learning is hebbian. *F1000Res. 2017 Jul 25;6:1222. doi: 10.12688/f1000research.12130.2. PMID: 29071065; PMCID: PMC5639933.*, 2017.

[41] Y. Shang, M. Cai, B. Xu, Y. J. Lee, and Y. Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models, 2026.

[42] X. Shen, Y. Xiong, C. Zhao, L. Wu, J. Chen, C. Zhu, Z. Liu, F. Xiao, B. Varadarajan, F. Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[43] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[44] X. Tang, J. Qiu, L. Xie, et al. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29118–29128, 2025.

[45] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

[46] I. Tenney, D. Das, and E. Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[47] C. Wang, X. Tan, J. Cao, K. Li, and T. Chen. CurveStream: Boosting streaming video understanding in MLLMs via curvature-aware hierarchical visual memory management. In *Computer Vision – ECCV 2026*, volume 17053 of *Lecture Notes in Computer Science*, pages 605–623. Springer, 2026.

[48] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[49] Y. Wang, Y. Song, C. Xie, et al. Videollamb: Long streaming video understanding with recurrent memory bridges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24170–24181, 2025.

[50] H. Xiong, Z. Yang, J. Yu, Y. Zhuge, L. Zhang, J. Zhu, and H. Lu. Streaming video understanding and multi-round interaction with memory-enhanced knowledge. *arXiv preprint arXiv:2501.13468*, 2025.

[51] B. Xu, Z. Xiao, J. Li, et al. Timeviper: A hybrid mamba-transformer vision-language model for efficient long video understanding. *arXiv preprint arXiv:2511.16595*, 2025.

[52] R. Xu, G. Xiao, Y. Chen, L. He, K. Peng, Y. Lu, and S. Han. Streamingvlm: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025.

[53] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 2023.

[54] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[55] P. Ye, T. He, B. Li, T. Chen, L. Bai, and W. Ouyang. β-DARTS++: Bi-level regularization for proxy-robust differentiable architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(2):1407–1423, 2026.

[56] P. Ye, T. He, S. Tang, B. Li, T. Chen, L. Bai, and W. Ouyang. Stimulative training++: Go beyond the performance limits of residual networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6463–6479, 2025.

[57] P. Ye, C. Huang, M. Shen, T. Chen, Y. Huang, and W. Ouyang. Dynamic model merging with mixture of weights. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(8):7925–7939, 2025.

[58] P. Ye, B. Li, Y. Li, T. Chen, J. Fan, and W. Ouyang. β-DARTS: Beta-decay regularization for differentiable architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10874–10883, 2022.

[59] X. Zeng, K. Qiu, Q. Zhang, X. Li, J. Wang, J. Li, Z. Yan, K. Tian, M. Tian, X. Zhao, et al. Streamforest: Efficient online video understanding with persistent event memory. *arXiv preprint arXiv:2509.24871*, 2025.

[60] H. Zhang, Y. Wang, Y. Tang, Y. Liu, J. Feng, J. Dai, and X. Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024.

[61] H. Zhang, S. Zhang, K. Li, C. Zhang, Y. Chen, Y. Zhang, L. Bai, and S. Hu. Self-Harness: Harnesses that improve themselves, 2026.

[62] P. Zhang, K. Zhang, B. Li, G. Zeng, J. Yang, Y. Zhang, Z. Wang, H. Tan, C. Li, and Z. Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[63] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[64] J. Zhou, Y. Shu, B. Zhao, B. Wu, Z. Liang, S. Xiao, M. Qin, X. Yang, Y. Xiong, B. Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13691–13701, 2025.

A Inference Efficiency and Memory Analysis

To quantitatively assess the computational and memory overhead introduced by our proposed FreshMem, we conducted rigorous inference time and memory usage measurements on the Qwen2-VL-7B model using the AVA subset from OV-Bench —a representative benchmark for streaming video understanding. Our evaluation focuses on absolute inference latency, peak GPU memory usage, and the corresponding performance gains. The results are summarized in Table 6.

Table 6: **Inference Efficiency Analysis on OV-Bench (AVA Subset).** Comparison of inference latency, peak GPU memory usage, and accuracy between the baseline and our proposed method. FreshMem achieves a significant performance boost (+1.79%) with a moderate and acceptable increase in computational overhead, ensuring deployment feasibility on consumer-grade GPUs (e.g., RTX 4090).

Method	Latency	GPU Mem.	Accuracy (%)
Qwen2-VL-7B	1h 21min	17GB	45.63
+ FreshMem (Ours)	1h 35min	20GB	**47.42**

Controlled Computational and Memory Overhead. Compared to the vanilla Qwen2-VL-7B baseline, FreshMem introduces a moderate and highly controlled increase in resource consumption. Specifically, the relative time overhead is approximately $\times 1.17$ (increasing from 81 mins to 95 mins), and the GPU memory usage sees a marginal increase of roughly 17.6% (from 17GB to 20GB). Crucially, the peak memory footprint of 20GB remains well within the capacity of standard consumer-grade GPUs (e.g., RTX 3090/4090), ensuring that our method maintains broad hardware accessibility and operational feasibility without requiring specialized high-end infrastructure.

Breakdown of Overhead Sources. The primary computational overhead in our method stems from the specific memory operations: the incremental Discrete Fourier Transform (DFT) updates in the MFM module and the centroid-based clustering in the STM module. However, these components are mathematically designed to be lightweight with near-linear complexity regarding the window size. By avoiding the heavy retraining of the entire model, FreshMem minimizes the computational burden typically associated with long-context adaptation.

Efficiency-Performance Trade-off. The marginal additional resource consumption is well justified by the substantial performance gains observed. While incurring only a slight overhead in latency and memory, FreshMem achieves a significant performance boost, improving accuracy from 45.63% to 47.42% ($+1.79\%$). This highlights that our approach strikes a highly favorable balance between computational complexity and memory-enhanced modeling capability. For latency-sensitive streaming applications, FreshMem offers a high-yield upgrade path that significantly enhances long-horizon reasoning with negligible impact on system throughput.

Summary. FreshMem maintains operational feasibility with predictable computational costs. It integrates seamlessly with the base model, providing a superior performance-efficiency trade-off that makes it suitable for both research exploration and real-world deployment scenarios where resources are constrained but high performance is demanded.

B Detailed Hyperparameter Settings

To verify the impact of key hyperparameters and determine the optimal configuration, we conduct a series of experiments on the OV-Bench Subset (AVA). The results are illustrated in Figure 6, covering sliding window length, memory module capacities (MFM and STM), frequency band, and the episode segmentation threshold θ_{event}. As shown in the top-left of Figure 6, performance peaks when the sliding window length is set to approximately 5 frames. A shorter window fails to capture sufficient temporal context, while an excessively long window introduces redundancy or noise, leading to performance degradation. Then we analyzed the capacity settings for MFM and STM modules in the top-middle and top-right plots of Figure 6. For MFM, performance reaches optima at capacities of 15. For STM, accuracy improves with capacity, saturating and achieving

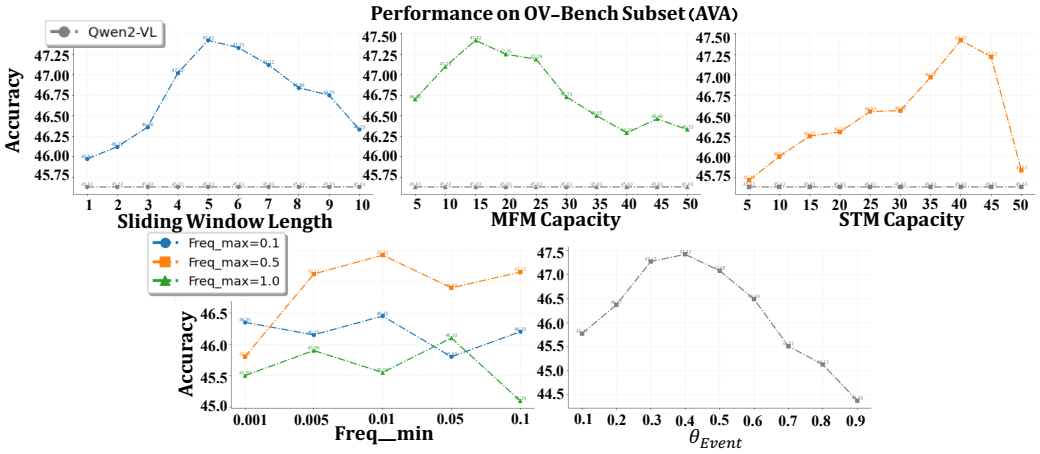


Figure 6: **Sensitivity analysis of FreshMem hyperparameters.** Experiments on the OV-Bench subset demonstrate the robustness of our default settings.

near-optimal performance around a capacity of 40. Considering the trade-off between inference GPU memory usage, computational latency, and model accuracy, we prioritized efficiency.

The bottom-left plot in Figure 6 illustrates the model's sensitivity to frequency band parameters ($Freq_min$ and $Freq_max$). The orange curve indicates that setting the high-frequency cutoff ($Freq_max$) to 0.5 yields the best overall performance. Furthermore, we observed that the model extracts the most discriminative temporal features when the high and low frequencies differ by approximately one order of magnitude.

Finally, we investigated the influence of the event update threshold, θ_{event}, within the STM module (Figure 6, bottom-right). This parameter controls the model's sensitivity in distinguishing different episodes. The data shows that the model achieves a stable peak performance when θ_{event} is set between 0.3 and 0.4. Values outside this range resulted in a notable drop in accuracy.

Table 7: **Hyperparameter settings in our experiments.**

Hyperparameter	**Value**
Sliding Window Length	5
MFM Capacity	15
STM Capacity	40
Number of Frequency Bands	16
Frequency Range	$0.01 \sim 0.5$
Decay Factor (γ)	0.9
Residual Tokens Ratio	Top 10%
Event Threshold (θ_{event})	0.4
Merge Threshold (θ_{merge})	0.3
Sampling Rate (ρ)	$1/16 \sim 1/4$

C Detailed Experimental Performance

We present the full evaluation results on OVO-Bench and MLVU in Tables 8 and 9. Table 8 shows the full evaluation results on OVO-Bench with all 12 subtasks. FreshMem demonstrates significant improvements over the Qwen2VL-7B baseline and outperforms many offline and online video MLLMs. Improvement on a specific subtask is up to 11.45%. Table 9 shows the full evaluation results on MLVU with all 7 MCQA subtasks. FreshMem outperforms baseline by 2.4% overall and up to 8.7% on a specific subtask.

Table 8: Full Evaluation results on OVO-Bench.

Model	Real-Time Visual Perception							Backward Tracing				Forward Active Responding				Overall Avg.
	OCR	ACR	ATR	STU	FPD	OJR	Avg.	EPM	ASI	HLD	Avg.	REC	SSR	CRR	Avg.	
Offline Video MLLMs																
Gemini 1.5 Pro [45]	85.91	66.97	79.31	58.43	63.37	61.96	69.32	58.59	76.35	52.64	62.54	35.53	74.24	61.67	57.15	63.00
GPT-4o [19]	69.80	64.22	71.55	51.12	70.30	59.78	64.46	57.91	75.68	48.66	60.75	27.58	73.21	59.40	53.40	59.54
Offline Video MLLMs																
LLaVA-Video-7B [63]	69.80	59.63	66.38	50.56	72.28	61.41	63.34	51.18	64.19	9.68	41.68	34.10	67.57	60.83	54.17	53.06
LLaVA-OneVision-7B [24]	67.11	58.72	69.83	49.44	71.29	60.33	62.79	52.53	58.78	23.66	44.99	24.79	66.93	60.83	**50.85**	52.88
InternVL2-8B [8]	68.46	58.72	68.97	44.94	67.33	55.98	60.73	43.10	61.49	27.41	44.00	25.79	57.55	52.92	45.42	50.05
LongVU-7B [42]	55.70	49.54	59.48	48.31	68.32	63.04	57.40	43.10	66.22	9.14	39.49	16.62	69.00	60.00	48.54	48.48
Online Video MLLMs																
VideoLLM-online-8B [7]	8.05	23.85	12.07	14.04	45.54	21.20	20.79	22.22	18.80	12.18	17.73	-	-	-	-	12.84
Flash-VStream-7B [60]	25.50	32.11	29.31	33.71	29.70	28.80	29.86	36.36	33.78	5.91	25.35	5.44	67.25	60.00	44.23	33.15
Dispider-7B [36]	57.72	49.54	62.07	44.94	61.39	51.63	54.55	48.48	55.41	4.30	36.06	18.05	37.36	48.75	34.72	41.78
Qwen2VL-7B [48]	69.13	53.21	63.79	50.56	66.34	60.87	60.65	45.79	48.65	54.84	49.13	30.09	65.66	50.83	48.86	52.19
+FreshMem	77.18	60.55	70.69	56.74	63.37	70.65	66.67	57.24	56.76	46.77	54.04	23.93	72.66	55.00	48.25	**54.53**(↑ 2.34%)

Table 9: Full Evaluation results on MLVU.

Method	TR	AR	NQA	ER	PQA	AO	AC	AVG
MovieChat [43]	18.7	10.3	23.3	15.1	16.0	17.1	15.0	16.5
MA-LMM [15]	44.0	23.1	13.3	30.2	14.0	18.6	13.3	22.4
LLaVa-OneVision-7B [24]	83.5	56.4	46.7	58.4	58.0	35.7	23.3	51.7
Qwen2VL-7B [48]	80.3	70.0	61.7	48.6	56.2	40.9	20.4	54.9
+FreshMem	83.0	73.5	72.4	53.1	53.6	43.6	16.5	**57.3**(↑ 2.4%)

D Additional Technical Analysis and Experimental Results

In this appendix, we provide further analysis of the memory representation and read/write operations in FreshMem. Unless otherwise specified, the additional experiments use Qwen2-VL-7B as the backbone.

D.1 Definition of the Spectral Frequencies

The frequency range $[0.01, 0.5]$ specifies *angular frequencies* in radians per sampled frame (rad/sample), not normalized ordinary frequencies or frequencies in radians per second. These values are directly used as ω_k, without an additional multiplication by 2π. In particular, t denotes the index of the sampled video frames rather than the original video-frame index or a wall-clock timestamp. The conversion to ordinary discrete frequency gives

$$f_k = \frac{\omega_k}{2\pi} \in [0.0016, 0.0796] \text{ cycles/sample}, \tag{10}$$

corresponding to periods

$$T_k = \frac{2\pi}{\omega_k} \in [12.6, 628.3] \text{ sampled frames}. \tag{11}$$

We intentionally retain a low-frequency subband: MFM aims to represent medium- and long-term temporal structures and coarse historical outlines, not to reconstruct every frame-to-frame variation losslessly. The sliding window and sparse residual pathway complement this representation with more recent and fine-grained visual information. Because the frequencies are defined in the sampled-frame domain, their physical-time meaning varies with the frame-sampling rate; we do not assume that the same frequency bins have identical physical-time interpretations across different FPS settings.

D.2 Budget-Matched Ablation of the Frequency and Residual Pathways

To isolate spectral reconstruction from simply retaining additional in-distribution visual embeddings, we compare residual-only, frequency-only, and full MFM variants under a matched long-term memory-token budget. These variants share the same backbone, video inputs and FPS, sliding-window configuration, and STM configuration. Their MFM-related output budgets are all 5,182 tokens. The no-MFM control retains the other memory components but does not allocate tokens to either MFM pathway; therefore, this control is not itself a token-budget-matched variant.

Model	Frequency	Residual	Total	Avg.
No MFM	–	–	–	53.92
Residual-only	0	5,182	5,182	54.07
Frequency-only	5,182	0	5,182	54.44
Full MFM	4,662	520	5,182	**54.53**

Table 10: Budget-matched MFM ablation on OVO-Bench (Qwen2-VL-7B). Frequency, residual, and total refer to the reported long-term MFM output-token budgets.

Question type	Baseline	MFM-only	STM-only	FreshMem
CS	73.19	77.21	74.50	**77.60**
EU	75.00	77.82	76.94	**78.12**
ACP	66.57	66.78	**66.94**	66.86
SU	65.04	70.85	74.10	**74.39**

Table 11: Contribution of MFM and STM across StreamingBench question types (Qwen2-VL-7B).

Table 10 shows that residual-only memory yields a limited improvement over the no-MFM control, whereas frequency-only memory obtains 54.44, exceeding residual-only (54.07) by 0.37 points under the same MFM token budget. The combined frequency-plus-residual design achieves the best result of 54.53. These findings indicate that the frequency-domain representation is the primary source of the measured gain in this comparison, rather than the mere presence of additional raw visual tokens. The smaller additional gain from the full model supports the complementarity of the two pathways: frequency coefficients preserve distributed global historical structure, and residual tokens compensate for details lost during spectral approximation.

D.3 Complementary Contributions of MFM and STM

We further investigate the different functions of MFM and STM across four StreamingBench question types: Clips Summarization (CS), Event Understanding (EU), Action Perception (ACP), and Spatial Understanding (SU). Here, *MFM-only* denotes a module-level configuration without STM, whereas *frequency-only* in Table 10 denotes removing the residual pathway within MFM.

As reported in Table 11, MFM contributes more strongly to long-range temporal questions such as CS and EU, whereas STM provides larger gains for spatial understanding and a small advantage in ACP. The complete FreshMem achieves the highest results on CS, EU, and SU; STM-only is slightly higher on ACP. These results support the functional complementarity of distributed spectral memory and episode-organized spatial evidence.

D.4 Selection of Salient Residual Tokens

The residual branch ranks *pre-LLM visual features* using their L_2 magnitudes rather than the decoder's key vectors after the W_K projection. Since these representations occupy different feature spaces, observations associating smaller norms of decoder keys with higher attention scores do not directly determine the appropriate selection rule for visual-encoder outputs. We therefore interpret the L_2 magnitude as a lightweight, magnitude-based intrinsic visual saliency score, rather than a universal proxy for information entropy. This distinction is also consistent with analyses of pre-LLM visual-token magnitudes in ST-GridPool.

Under an identical residual-token budget, the highest-norm strategy achieves 54.53, compared with 53.08 for random selection and 52.43 for lowest-norm selection (Table 12). These controlled results support high-norm selection in the visual feature space used by FreshMem, without assuming that the same criterion applies to all representations or attention mechanisms.

D.5 Robustness to Frame-Sampling Rate

Because ω_k is defined over the sampled-frame index, different sampling rates alter the physical timescales represented by the fixed frequency bins. To investigate the resulting sensitivity, we vary the sampling rate from $0.25\times$ to $4\times$ on OVO-Bench. For all settings, we keep the angular-frequency

Residual selection strategy	Score
Baseline	52.19
Lowest-norm tokens	52.43
Random tokens	53.08
Highest-norm tokens	**54.53**

Table 12: Residual-selection strategies on OVO-Bench (Qwen2-VL-7B), evaluated with the same residual-token budget.

Sampling rate	Baseline	FreshMem	Improvement
$0.25\times$ FPS	49.06	52.94	+3.88
$0.5\times$ FPS	51.42	54.60	+3.18
$1\times$ FPS	52.19	54.53	+2.34
$2\times$ FPS	53.72	56.25	+2.53
$4\times$ FPS	52.30	54.77	+2.47

Table 13: Robustness to different video-frame sampling rates on OVO-Bench (Qwen2-VL-7B).

range and number of frequency bands, the memory-token budget, the sliding-window and STM configurations, and the decoding configuration fixed.

As shown in Table 13, changing the sampling rate affects absolute performance for both the baseline and FreshMem, but FreshMem provides positive improvements relative to the corresponding baseline at every tested rate. Thus, the observed benefit does not depend on a single FPS. Nevertheless, low sampling rates or irregular sampling may introduce aliasing or phase distortion for rapid events. Defining frequencies in rad/s using actual timestamps is a possible extension for improving consistency of the physical-time interpretation across FPS.

D.6 Sensitivity to the Number of Frequency Bands and Decay

We evaluate the number of frequency bands K and decay factor γ on the AVA subset of OV-Bench. Both experiments use a reported baseline score of 45.50. Table 14 shows that increasing K from 4 to 16 improves performance from 44.82 to 47.60; subsequent increases yield only small variations. We therefore use $K = 16$ as a practical trade-off between accuracy and inference cost.

The decay study in Table 15 shows that performance peaks at $\gamma = 0.90$ among the evaluated settings and remains close at $\gamma = 0.85$ and 0.95. A smaller γ attenuates distant historical information more rapidly, while larger values can increase interference from older contexts. Extreme settings, including $K = 4$ and $\gamma = 0.70$, can underperform the baseline. The additional hyperparameter analysis in the original appendix also evaluates sliding window length, memory capacities, and STM event thresholds; the event threshold θ_{event} exhibits stable performance over approximately $[0.3, 0.5]$, and MFM/STM capacities are examined over 5–50. The same default hyperparameters are used across benchmarks and backbones, without dataset-specific retuning.

D.7 Additional Results after Component-Wise Fine-Tuning

We additionally evaluate the component-wise fine-tuned configurations on OV-Bench and OVO-Bench, extending the StreamingBench fine-tuning results in the main paper. The newly evaluated fine-tuned settings follow the end-to-end training configuration of StreamForest. The training-free FreshMem baseline is included to separate the effect of memory integration from the additional benefits of fine-tuning.

As reported in Table 16, all three fine-tuned configurations improve over training-free FreshMem. In particular, LLM fine-tuning reaches 62.40 on OV-Bench and 58.81 on OVO-Bench, exceeding the reported StreamForest results of 60.50 and 55.57, respectively. These results complement the StreamingBench fine-tuning comparisons and further demonstrate the compatibility of FreshMem with component-wise adaptation.

K	Base	4	8	16	24	32	64
Score	45.50	44.82	46.58	**47.60**	47.55	47.63	47.58

Table 14: Sensitivity to the number of frequency bands K on the AVA subset of OV-Bench. Base denotes the Qwen2-VL-7B baseline.

γ	Base	0.70	0.80	0.85	0.90	0.95	0.99
Score	45.50	44.75	46.20	47.53	**47.60**	47.58	46.81

Table 15: Sensitivity to the decay factor γ on the AVA subset of OV-Bench.

D.8 Task-Dependent Behavior and Lossy Reconstruction

MFM synthesizes an approximate historical *gist* rather than an exact archive of all past visual tokens. This representation is designed for distributed long-range temporal structure and may be less effective when a question requires exact textual content (e.g., OCR or subtitles), small objects or fine-grained attributes (e.g., colors and counts), or transient events lasting only one or two frames. The hybrid design addresses different information needs through the recent high-fidelity sliding window, sparse salient residuals, and episode-level STM evidence.

The question-type results in Table 17 illustrate this trade-off. The frequency-only branch provides larger improvements for CS and EU, while the residual-only branch is relatively more helpful than the frequency-only branch for Counting (CT). The full MFM variant obtains the best scores for CS and EU, but CT remains below the baseline even after combining both pathways. These observations support the complementary roles of the two pathways while identifying fine-grained counting as a limitation of lossy frequency reconstruction.

D.9 Long-Stream Scene Transitions and Concept Drift

Extremely long streams may undergo persistent shifts in scene and semantic distribution. Expanding the recurrent MFM update gives

$$C_t[k] = \gamma^t C_0[k] + \sum_{i=1}^{t} \gamma^{t-i} x_i e^{-j\omega_k i}. \tag{12}$$

Thus, the contribution of historical feature x_i is attenuated by γ^{t-i} as the stream advances. Exponential decay gradually reduces the influence of old scenes; in parallel, the sliding window provides immediate raw visual context after a scene transition.

We further assess scene-transition sensitivity by concatenating unrelated videos and evaluating on the Causal Reasoning (CR) subset of StreamingBench. As shown in Table 18, the score changes from 83.59 before the transition to 80.21 immediately afterward, then recovers to 82.73 after several update steps. This result is consistent with the decay mechanism and recent-window features mitigating interference from obsolete context. However, a fixed γ may not be optimal under slow or persistent distribution shifts; adaptive decay and boundary-triggered memory reset or gating remain possible future extensions, rather than mechanisms already established by this model.

E Additional Related Work

Memory and Long-Horizon Adaptation. Recent studies have explored various mechanisms for improving long-context modeling and adaptive memory management. PaceLLM [27] introduces persistent activation memory and cortical expert clustering to enhance long-context understanding in language models. CurveStream [47] employs curvature-aware hierarchical visual memory management to preserve informative frames and semantic transitions in streaming videos. Beyond inference-time memory organization, ReTTT [28] incorporates reflection-driven experience into parametric memory through test-time training, while Self-Harness [61] enables agents to iteratively improve their execution harnesses based on observed failures. These approaches investigate complementary mechanisms for information retention, adaptation, and long-horizon reasoning.

Method	OV-Bench	OVO-Bench
VideoChat-Online	54.90	–
StreamForest	60.50	55.57
Qwen2-VL-7B	46.30	52.19
+ FreshMem (training-free)	50.82	54.53
+ FreshMem (ViT fine-tuned)	56.49	55.24
+ FreshMem (Projector fine-tuned)	59.72	57.10
+ FreshMem (LLM fine-tuned)	**62.40**	**58.81**

Table 16: Additional comparisons on OV-Bench and OVO-Bench before and after component-wise fine-tuning. A dash denotes an unavailable reported result.

Question type	Baseline	Residual-only	Frequency-only	Full MFM
CT	**35.75**	32.59	29.16	30.05
CS	73.19	73.32	76.38	**77.60**
EU	75.00	76.38	77.86	**78.12**

Table 17: Question-type analysis of the frequency and residual branches on StreamingBench (Qwen2-VL-7B). CT, CS, and EU denote Counting, Clips Summarization, and Event Understanding.

Optimization and Model Adaptation. Related advances in neural network optimization and model composition also provide broader perspectives on adaptive model design. Beta-DARTS [58] introduces beta-decay regularization to stabilize differentiable architecture search, while Beta-DARTS++ [55] further improves proxy robustness through bi-level regularization. Stimulative Training++ [56] addresses network loafing in residual networks through subnetwork supervision and improved training strategies. Dynamic Model Merging with Mixture of Weights [57] introduces input-dependent weight merging to reduce interference across task-specific models. Although these methods focus on optimization and model composition rather than streaming video memory, they offer complementary perspectives on efficient and adaptive neural network design.

F More Experiments on Other Models.

Table 19: **More Experiments on InternVL3.5-7B and DeepSeek-VL2-Tiny.**

Method	StreamingBench	OV-Bench	OVO-Bench
InternVL3.5-7B	75.26	53.33	56.49
+FreshMem	**80.02**(↑ 4.76%)	**58.78**(↑ 5.45%)	**58.81**(↑ 2.32%)
DeepSeek-VL2-Tiny	53.32	44.57	43.23
+FreshMem	**58.39**(↑ 5.07%)	**49.25**(↑ 4.68%)	**45.18**(↑ 1.95%)

G More Visualization Results

Figures 7 to 10 show more video frames of the current streaming video reconstructed from the MFM modules based on different DFT coefficients at different steps. We can still observe that MFM is capable of retaining the gist information from the past history, which is consistent with our discussion in the main text. Figures 11 to 15 show more residual tokens reconstructed in the MFM module. From the results, it can be seen that the crucial residual information is basically concentrated in the semantic important areas, which is consistent with our main conclusion in the text. Importantly, even for video material such as that shown in Figure 15—where the scene is very dark yet full of crucial dialogues or subtle actions—the semantic information can still be effectively captured. MFM can maintain the historical semantic outline while preserving high-fidelity visual information.

Setting	Score
Baseline	76.56
Before the scene transition	83.59
Immediately after the transition	80.21
After several update steps	82.73

Table 18: Stress test with concatenated unrelated videos on the Causal Reasoning (CR) subset of StreamingBench (Qwen2-VL-7B).

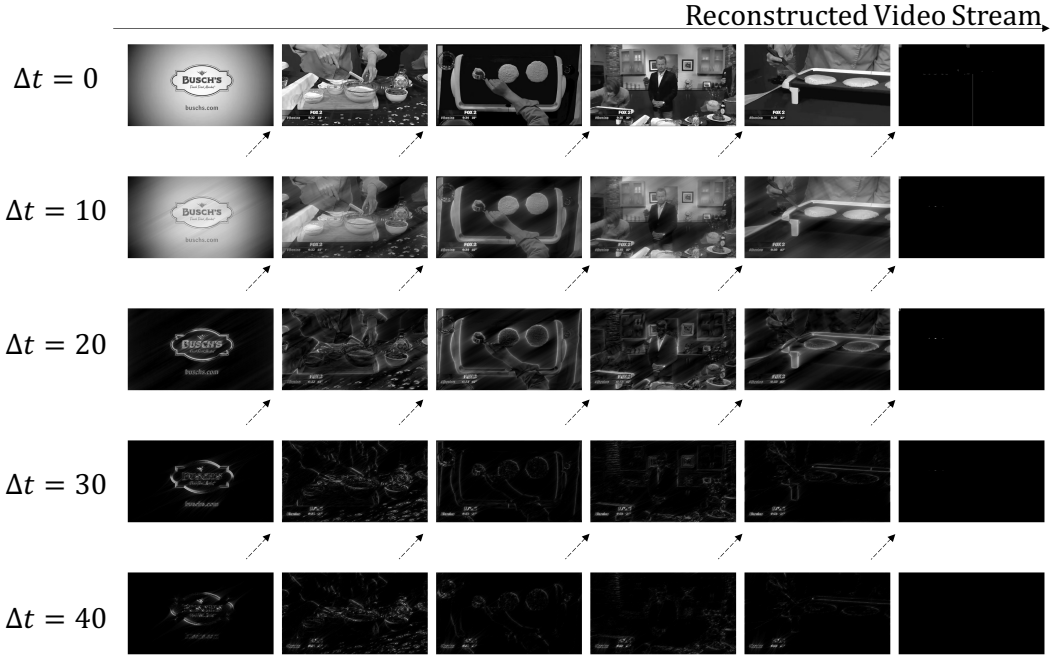


Figure 7: **Reconstructed video stream from frequency coefficients.**

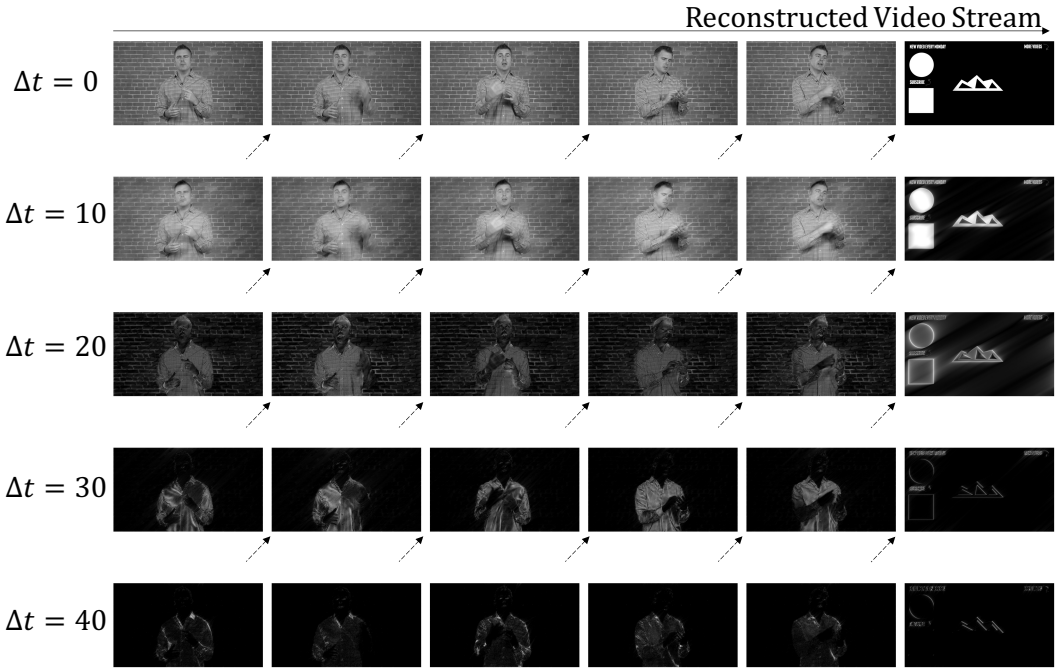


Figure 8: **Reconstructed video stream from frequency coefficients.**

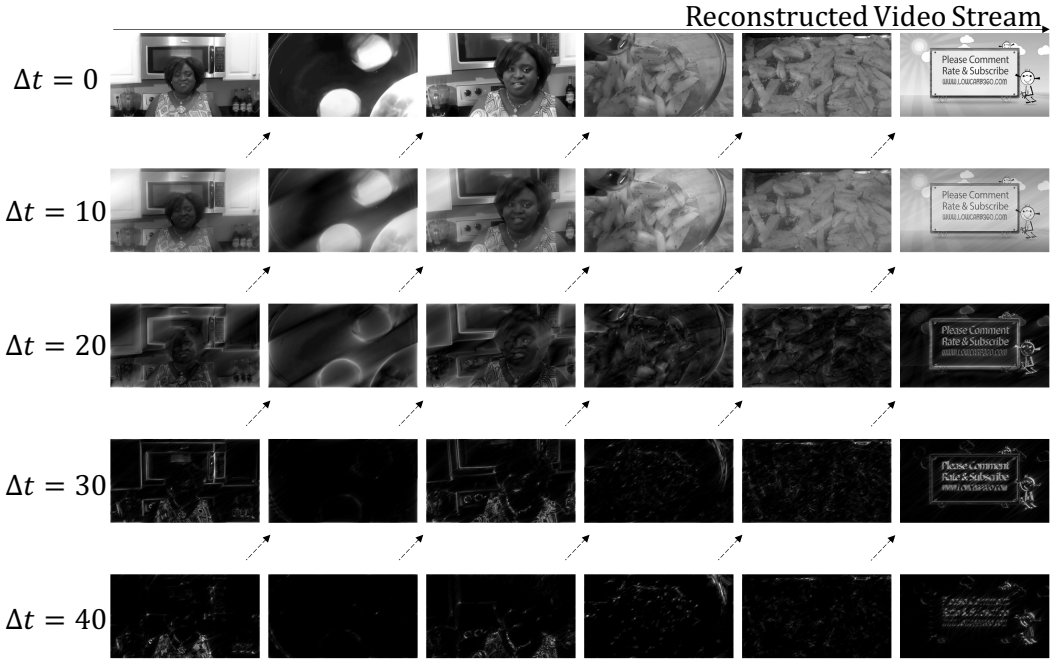


Figure 9: **Reconstructed video stream from frequency coefficients.**

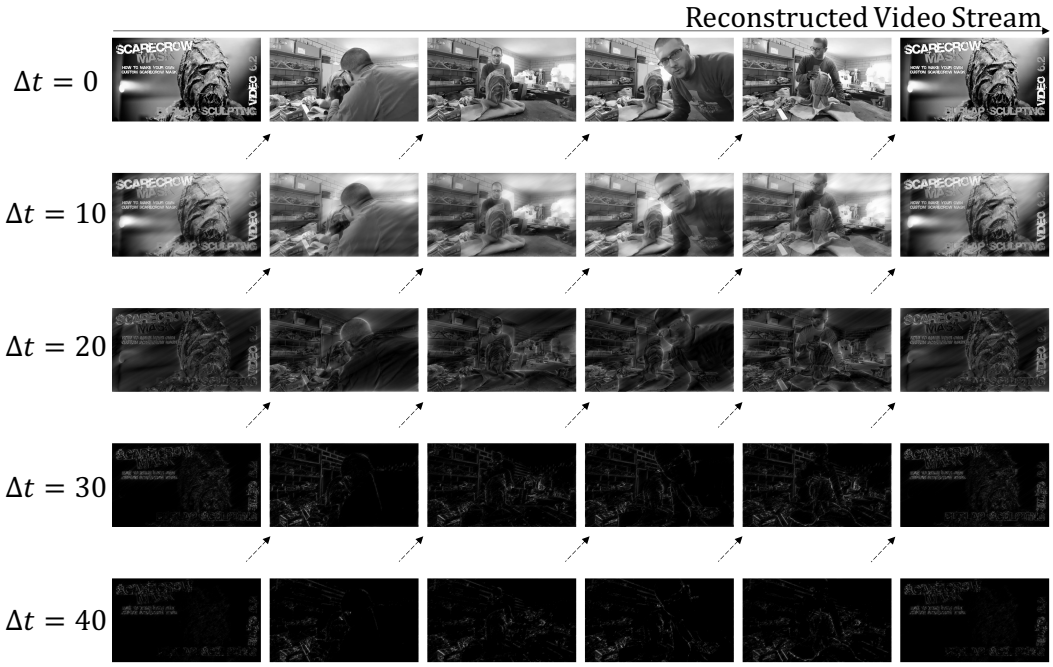


Figure 10: **Reconstructed video stream from frequency coefficients.**

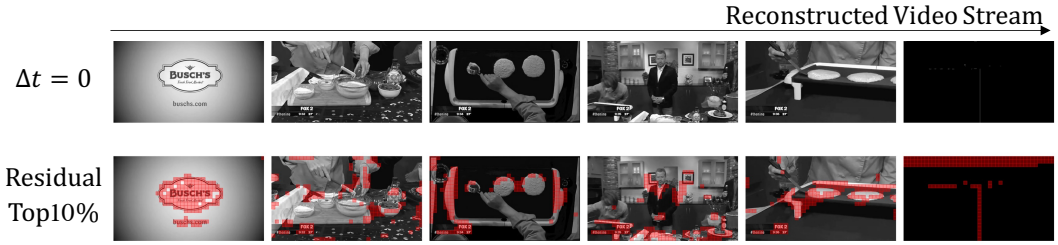


Figure 11: **Impact of Salient Residual Tokens.**

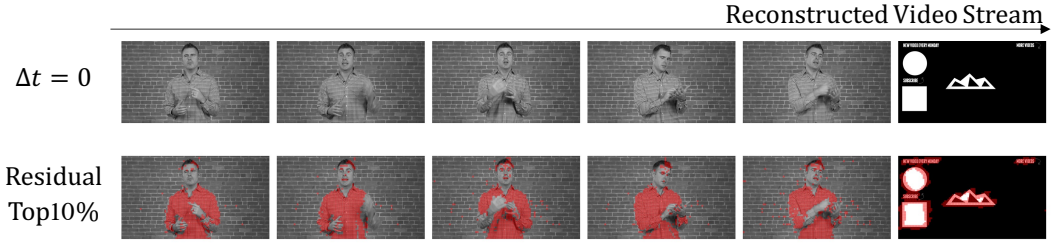


Figure 12: **Impact of Salient Residual Tokens.**



Figure 13: **Impact of Salient Residual Tokens.**

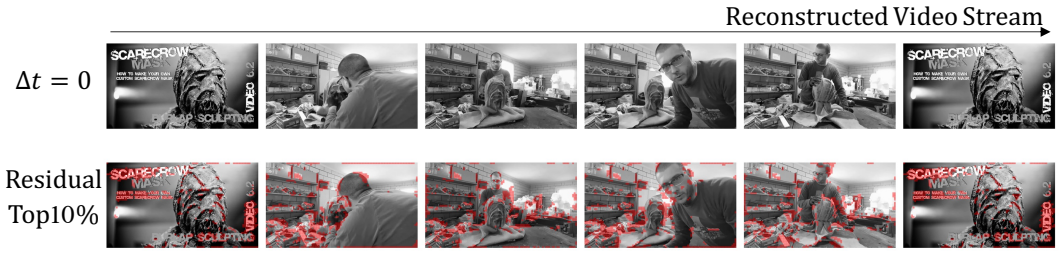


Figure 14: **Impact of Salient Residual Tokens.**

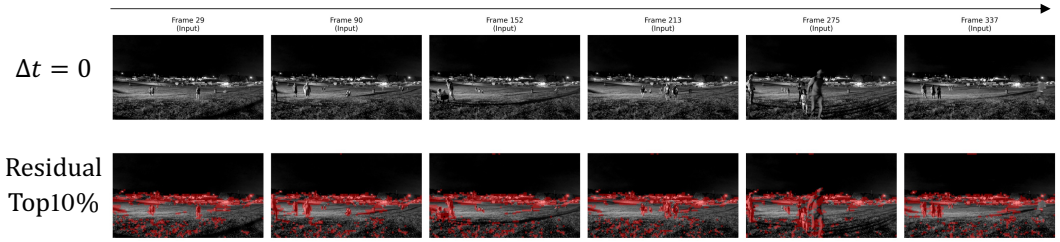
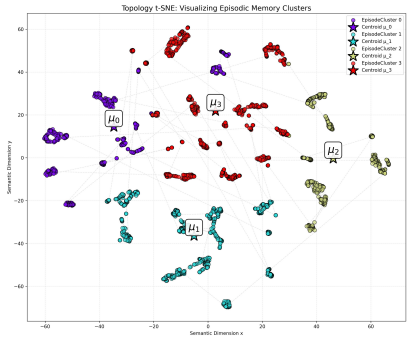


Figure 15: **Impact of Salient Residual Tokens.**

H More Case Studies

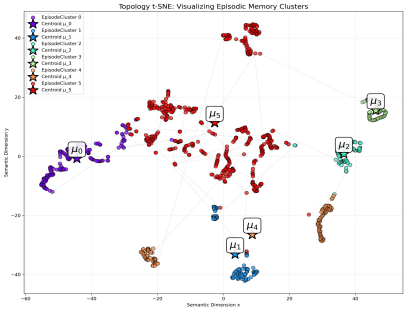




Question: Early in the video, while walking through the produce section, the camera captured a man wearing a black jacket and sunglasses. What specific color of apples was he reaching for?
Options:
A:Red
B:Green
C:Yellow
D:He was reaching for oranges
Answer: B **Model Prediction:B** ✓

Figure 16: **Interpretability analysis of the STM module.**

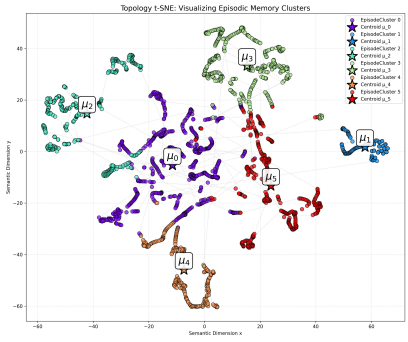




Question: After getting out of the car and walking towards the house, what object is the character (Mr. Bean) carrying in his hand?
Options:
A:952R
B:STE 952R
C:SLW 287R
D:BEAN 01
Answer: B **Model Prediction:B** ✓

Figure 17: **Interpretability analysis of the STM module.**





Question: During the frying and assembly stages, a specific pair of tongs is placed on the stovetop and later used to move ingredients. What is the color of these tongs?
Options:
A:Silver
B:Black
C:Magenta
D:Blue
Answer: C **Model Prediction:C**

Figure 18: **Interpretability analysis of the STM module.**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All major claims stated in the abstract and introduction are thoroughly substantiated by the results and analyses presented in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our work's limitations are discussed in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: No theoretical results are included in this paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All of the necessary details to make our work reproducible are included in this paper. Our methodology is described in detail in Section 3, and our experimental setup and implementation details are included in Section 4. Additionally, all of our code and data will be included in the submission and released upon acceptance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All of the code and data used in this study as well as the necessary documentation to run it will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: All of the necessary details for testing in terms of experimental setup and implementation details, including training splits and hyperparameter tuning can be found in Section 4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our method is inherently training-free, meaning it does not involve stochastic training processes (e.g., random initialization or data shuffling) that typically require multiple runs to assess variability. Since the approach operates deterministically on fixed pretrained models, identical inputs and configurations, as demonstrated in Section 4, will always produce the same outputs. Results are reproducible across identical hardware and software environments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the local computing resources we utilize in Section 4 and detailed time and costs will be put in Section A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the NeurIPS Code of Ethics and made sure that our paper conforms to it in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Our work promotes interdisciplinary research between AI and neuroscience. This may positively impact future research on interpretable and cognitively aligned AI. As a methodological contribution, it poses no foreseeable negative societal risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We release no models and the data we release is either already publicly available or purely the output of an LLM doing OpenIE on such data. We believe that this paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit the owners of all code, models and data used in this work. Much of this information can be found in Section 4.

    Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All of the code and data assets released alongside our paper are appropriately documented for reproducibility.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper involves no crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: Our paper involves no crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLMs used, such as ChatGPT, are limited to slight polishing and grammatical corrections of the paper language. They had no role in the design of core research methods, data analysis, experimental procedures, or interpretation of results. Therefore, the use of LLMs did not have an impact on the scientific rigor, originality or substance of the study, so it was not necessary to make a formal declaration in the paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.